

# Automatizovaná tvorba dokumentace

SPŠ Třebíč

Tomas FRYZA

Department of Radio Electronics, Brno University of Technology, Czechia  
Wed, 22 March 2023



## Contents

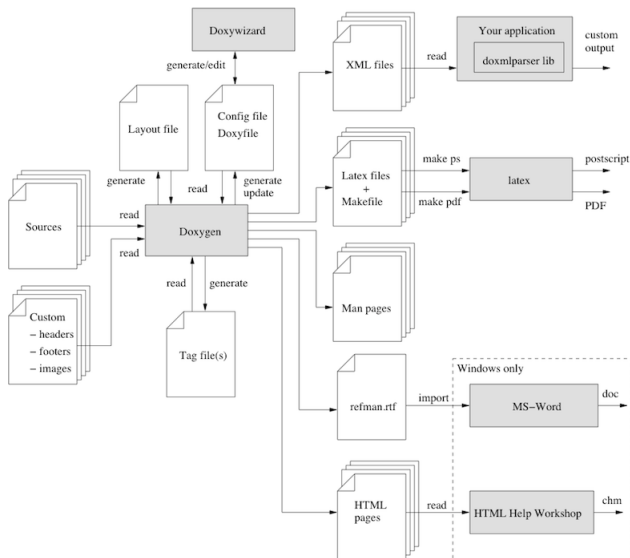
**1** How to generate documentation

**2** Contact information

# How to generate documentation

- 1 How to generate documentation
- 2 Contact information

- Doxygen ([www.doxygen.org](http://www.doxygen.org)): free, multiplatform (Linux, Windows, Mac, ...) tool for easy generation of program manual
- Main supported programming languages: C, C++, C#, PHP, Java, Python, Fortran, VHDL, and others
- Common generated outputs: HTML (mostly),  $\text{\LaTeX}$ , RTF, ...
- Fundamental: usage of several Doxygen keywords within your block comments:
  - for C, block comments have to start by two asterisks: `/** ... */` instead of `/* ... */`
  - in Python, common docstrings could be used (triple quotation marks at the beginning of class, funktion, ...) or two hash symbols: `## ...` instead of `# ...`
  - in VHDL `--!` instead of `--`
  - <https://www.doxygen.nl/manual/features.html>



- Except of generation tools, Doxygen has a configuration wizard (Doxywizard) and a tool for reading extract data from external HTML documentation (Doxytag)

- **Doxywizard** is GUI for setting all parameters of generation process. Output of the wizard is a text config file (Doxyfile)

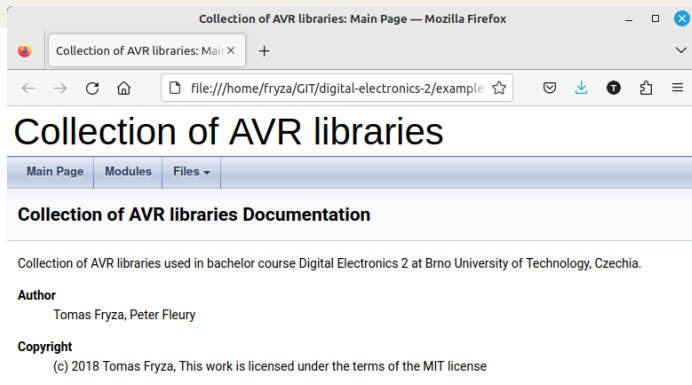
- **Doxytag** is a command line tool to extract information about source files, classes, functions (etc.) directly from existing HTML documentation. Therefore, the Doxygen input is not only the source files

L<sup>A</sup>T<sub>E</sub>X is external document preparation system

- In block comments, there are special Doxygen commands to specify generation
- Commands start by backslash "`\`" or by "`@`"
- Complete list of all commands:  
<https://www.doxygen.nl/manual/commands.html>

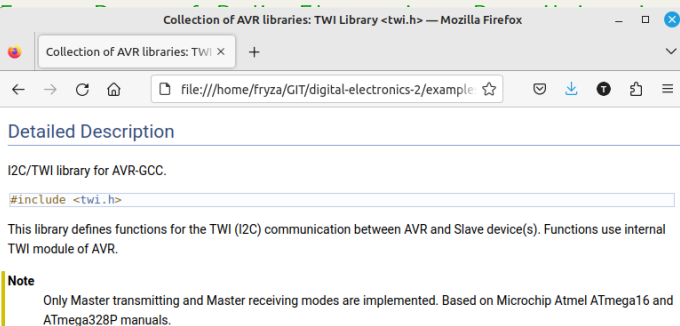
| Name      | Description                                  |
|-----------|--|
| @author   | Application/file author                      |
| @brief    | Brief description of code entity             |
| @date     | Date   |
| @details  | Detail description of code entity            |
| @mainpage | Content of main page <code>index.html</code> |
| @param    | Function parameter                           |
| @remarks  | Notes  |
| @return   | Return value                                 |
| @todo     | TODO section, i.e. what should be done       |
| @version  | Version of your documentation                |

```
/**
 * @mainpage
 *
 * Collection of AVR libraries used in bachelor course Digital Electronics 2 at Brno
 * University of Technology, Czechia.
 *
 * @author Tomas Fryza, Peter Fleury
 * @copyright (c) 2018 Tomas Fryza, This work is licensed under the terms of the MIT
 * license
 */
```



# Source module documentation (could be \*.c or \*.h)

```
/**
 * @file
 * @defgroup fryza_twi TWI Library <twi.h>
 * @code #include <twi.h> @endcode
 *
 * @brief I2C/TWI library for AVR-GCC.
 *
 * This library defines functions for the TWI (I2C) communication between
 * AVR and Slave device(s). Functions use internal TWI module of AVR.
 *
 * @note Only Master transmitting and Master receiving modes are implemented. Based
 * on Microchip Atmel ATmega16 and ATmega328P manuals.
 * @author Tomas Fryza, Faculty of Electrical Engineering and Communication,
 * of Technical University of Ostrava
 * @copyright (c) 2013 Tomas Fryza
 *
 * @{
 */
```



# Example of C-code defines

```
/**
 * @name Definition of frequencies
 */
#ifndef F_CPU
#define F_CPU 16000000 /**< @brief CPU frequency in Hz required TWI_BIT_RATE_REG */
#endif
#define F_SCL 50000 /**< @brief I2C/TWI bit rate. Must be greater than 31000 */
#define TWI_BIT_RATE_REG ((F_CPU/F_SCL - 16) / 2) /**< @brief TWI bit rate register
value */
```

Collection of AVR libraries: TWI Library <twi.h> — Mozilla Firefox

Collection of AVR libraries: TWI × +

file:///home/fryza/GIT/digital-electronics-2/example ☆

## Definition of frequencies

|  |  |
|--|--|
| <b>#define F_CPU</b> 16000000                            | CPU frequency in Hz required TWI_BIT_RATE_REG. |
| <b>#define F_SCL</b> 50000                               | I2C/TWI bit rate. Must be greater than 31000.  |
| <b>#define TWI_BIT_RATE_REG</b> ((F_CPU/F_SCL - 16) / 2) | TWI bit rate register value.                   |

## Definition of ports and pins

|                               |                   |
|-------------------------------|-------------------|
| <b>#define TWI_PORT</b> PORTC | Port of TWI unit. |
|-------------------------------|-------------------|

8 / 13



# Example of C-code function

```
/**
 * @brief Read one byte from I2C/TWI Slave device and acknowledge
 *        it by ACK or NACK.
 * @param ack — ACK/NACK value to be transmitted
 * @return Received data byte
 */
uint8_t twi_read(uint8_t ack);
```



◆ twi\_read()

uint8\_t twi\_read ( uint8\_t ack )

Read one byte from I2C/TWI Slave device and acknowledge it by ACK or NACK.

## Parameters

**ack** - ACK/NACK value to be transmitted

## Returns

Received data byte

◆ twi\_start()

# Generate it!

- Process of generation is controlled by config file with hundreds of parameters, see: <https://www.doxygen.nl/manual/config.html>
- Default config file Doxyfile with about 2 500 lines (see the next page) could be generated by GUI
- Config file Doxyfile could be edited in any text editor or GUI Doxywizard (preferred).
- Note, output layout could be modified in template DoxygenLayout.xml
- Basic settings of wizard is here: [https://www.doxygen.nl/manual/doxywizard\\_usage.html](https://www.doxygen.nl/manual/doxywizard_usage.html)

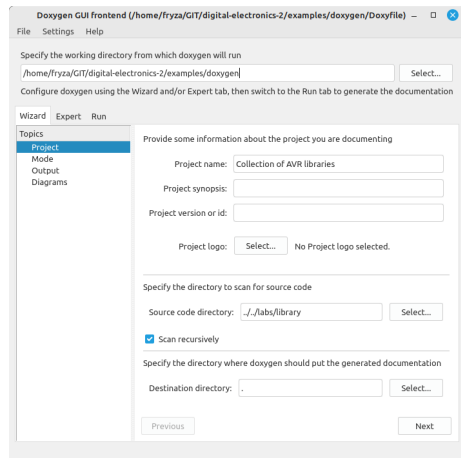


Figure 1.1: To edit config file and generate output(s), use the GUI Doxywizard

# Example of config file Doxyfile

```
# Doxyfile 1.8.17

# This file describes the settings to be used by the documentation system
# doxygen (www.doxygen.org) for a project.

...

# The default value is: UTF-8.

DOXYFILE_ENCODING      = UTF-8

# The PROJECT_NAME tag is a single word (or a sequence of words surrounded
# by
# double-quotes, unless you are using Doxywizard) that should identify the
# project for which the documentation is generated. This name is used in
# the
# title of most generated pages and in a few other places.
# The default value is: My Project.

PROJECT_NAME           = "Collection of AVR libraries"

# The PROJECT_NUMBER tag can be used to enter a project or revision number
# . This
# could be handy for archiving the generated documentation or if some
# version
# control system is used.
```

# Contact information

- 1 How to generate documentation
- 2 Contact information**



## Tomas FRYZA

Department of Radio Electronics  
Brno University of Technology  
[https://www.urel.fekt.vut.cz/  
bakalarsky-program](https://www.urel.fekt.vut.cz/bakalarsky-program)

[fryza@vut.cz](mailto:fryza@vut.cz)

**f** <https://www.facebook.com/URELBrno>

**g** <https://www.github.com/tomas-fryza>

**t** @TomasFryza